SOFTWARE ENGINEERING PROCESSES AND MANAGEMENT

Mike, Carmel, you go to the garden store and get five bags of lime. Dana, Gord, Jill, get me a tarp and carpet, then check Google Earth for a secluded stretch of highway. The rest of you, you'll help me bury the facilitator... who, ironically, has apparently just led the most successful team-building exercise in our company's history.

WORKSHOP 3
MEET THE TEAM

INTRODUCTION

The aim of this workshop is to reinforce the skills that you picked up in the previous workshop on designing lifecycles models, to get experience at designing team structures, and to explore the relationships that exist between lifecycle models and team structures.

The relationships between team structures and lifecycle models constrains the selection of both. This is both a benefit (there are less options to consider) and a hindrance (we may be constrained from using a particular solution).

TEAM STRUCTURES

Selecting a team structure has a large influence on projects. For example, do we want the client to be tightly integrated with the requirements engineers? On the one hand, it may mean that the client is engaged in the project closely, and can get what they want in less iterations. On the other hand, the client may unnecessarily hinder the progress of the requirements engineering team by focusing on small details, or asking irrelevant questions.

The following are some tips that may help in designing a team structure:

- Our team = good and productive; other teams = nuisance. This is often the view of many people: our team is doing valuable and important work, and the members are efficient and productive, while other teams are just a nuisance that hinder our progress. People in the same team often feel as though they are on the same side. Therefore, it is beneficial if teams are able to see the work being done in other teams on the same project, and to make everybody feel like one large team.

- Where possible, allow teams to remain together for as long as possible over the course of a project. It takes time for teams to learn to work with each other, therefore, minimising the number of new team formations is likely to be more efficient, as long as it does not detriment other areas of the project.

- You should be able to feed your teams with two pizzas. That is, keep the size of the teams to about 4-8 people. Larger teams seem to be far less efficient, and take much longer to make decisions, due to the many differing viewpoints. Furthermore, the larger the team, the more communication paths there are between members, and the easier it is for people to hide and do little work.

- Balance the communication paths between teams. Similar to the above point, the number of communication paths between teams must be large enough that all teams can get the information they require, while at the same time, must be small enough so as to not require one team to coordinate with several others just to confirm a simple design decision.

- Placing people with different skills in the same team encourages skill and knowledge transfer.

DESIGNING A SDLC AND TEAM STRUCTURE FOR A PROJECT

Your task in this workshop is to design a lifecycle process and a team structure for the language technology project.

Choose a software development lifecycle model (10 mins). The first task is to choose a lifecycle model for the language research case study project. This task you did last week, however, this time, take into account the tips above for forming teams, and discuss if you want to make changes to last week's solution.

Recall from the workshop last week that you will need to:

- Determine the key characteristics and goals for this project and work out which of the lifecycle models best fits the goals of the project.
- Sketch what process you would put in place to achieve the goals of each phase of your chosen lifecycle process.
- Justify your choice of lifecycle model based on you criteria.

Building a team (10 mins) Consider your own strengths and weaknesses, and the roles that you may play in a team working on this project. The roles come from Table 4.2 in the lecture notes handout — be honest now!

Design a team structure (15 mins). Design a team structure that fits within your software development lifecycle. For this task, you will need to identify the key roles that will be required in the project, the goals towards which these roles will be contributing, and the structure of the team itself.

Justify your process and team structure (10 mins). Nominate a member from your group to justify your process and team structure to the rest of the class. What particular aspects of the project lead you to choosing this combination of process and team structure?